https://wokwi.com/projects/413733657687500801

```
#include <DHT.h>

#include <Wire.h>

#include <LiquidCrystal_I2C.h>

#include "RTClib.h"


//PIN SENSOR TEMPERATURA

#define DHTPIN 2

#define DHTTYPE DHT22

// PIN SENSOR MOVIMIENTO

const int pirPin = 4;

// PIN BOTON RESETEO

const int buttonPin = 5;


// INICIALIZAR SENSOR

DHT dht(DHTPIN, DHTTYPE);


//CONTEO DE APERTURA DE PUERTA

int contador_p = 0;


//ESTADOS COMPOST

enum State {

  STATE_VACIO,

  STATE_LLENO,

  STATE_MESOFILA,

  STATE_TERMOFILA,

  STATE_FINALIZADO,

};
```

```
State state = STATE_VACIO;

// INICIALIZAR LCD (DISPLAY)
LiquidCrystal_I2C lcd(0x27, 20, 4);

// Inicializar RTC
RTC_DS1307 rtc;
DateTime startTime;

void aireando() {
  int i = 0;
  while (i != 3) {
    lcd.setCursor(0, 0);
    lcd.print("AIREANDO.          ");
    delay(1000);
    lcd.setCursor(0, 0);
    lcd.print("AIREANDO..         ");
    delay(1000);
    lcd.setCursor(0, 0);
    lcd.print("AIREANDO...        ");
    delay(1000);
    i++;
  }
  lcd.setCursor(0, 0);
  lcd.print("                    "); // Limpia la línea 0 al finalizar
}

void llenando() {
  int i = 0;
```

```
  while (i != 3) {
    lcd.setCursor(0, 0);
    lcd.print("LLENANDO.         ");
    delay(1000);
    lcd.setCursor(0, 0);
    lcd.print("LLENANDO..        ");
    delay(1000);
    lcd.setCursor(0, 0);
    lcd.print("LLENANDO...       ");
    delay(1000);
    i++;
  }
  lcd.setCursor(0, 0);
  lcd.print("                    "); // Limpia la línea 0 al finalizar
}

void setup() {
  Serial.begin(9600);

  // INICIALIZAR SENSOR
  dht.begin();
  // SENSOR MOVIMIENTO COMO ENTRADA
  pinMode(pirPin, INPUT);
  // BOTON DE RESETEO COMO ENTRADA
  pinMode(buttonPin, INPUT);

  // INICIALIZAR DISPLAY
  lcd.init();
  lcd.backlight();
```

```
  // INICIALIZAR RTC
  if (!rtc.begin()) {
    Serial.println("No se encontró el RTC");
    while (1);
  }

  if (!rtc.isrunning()) {
    Serial.println("ajustando la hora...");
    rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
  }

  // REGISTRAR EL TIEMPO INICIAL
  startTime = rtc.now();
}

void loop() {
  float temperature = dht.readTemperature();
  DateTime currentTime = rtc.now();

  // Muestra la temperatura y el conteo en las líneas 2 y 3
  lcd.setCursor(0, 2);
  lcd.print("Temperatura: ");
  lcd.print(temperature, 1);
  lcd.print("C");

  lcd.setCursor(0, 3);
  lcd.print("Conteo aireado: ");
  lcd.print(contador_p);
```

```
  switch (state) {
    case STATE_VACIO:
      lcd.setCursor(0, 0);
      lcd.print("                ");
      lcd.setCursor(0, 1);
      lcd.print("VACIO           ");

      if (digitalRead(pirPin) == HIGH) {
        llenando();
        startTime = currentTime;  // Guardar el tiempo de cambio de
estado
        state = STATE_LLENO;
      }
      break;

    case STATE_LLENO:
      lcd.setCursor(0, 0);
      lcd.print("                ");
      lcd.setCursor(0, 1);
      lcd.print("LLENO           ");

      if (digitalRead(buttonPin) == HIGH) {
        state = STATE_VACIO;
      }

      if ((currentTime - startTime).totalseconds() >= 15 && temperature
> 14) {
        startTime = currentTime;  // Actualizar tiempo para el siguiente
estado
```

```
    state = STATE_MESOFILA;
  }
  break;

case STATE_MESOFILA:
  lcd.setCursor(0, 1);
  lcd.print("ELEVANDO TEMPERATURA");

  if (digitalRead(buttonPin) == HIGH) {
    state = STATE_VACIO;
  }

  if (digitalRead(pirPin) == HIGH) {
    contador_p++;
    aireando();
  }

  if ((currentTime - startTime).totalseconds() >= 15 && temperature
> 40) {
    startTime = currentTime;
    state = STATE_TERMOFILA;
  }
  break;

case STATE_TERMOFILA:
  lcd.setCursor(0, 1);
  lcd.print("MAX TEMPERATURA      ");

  if (digitalRead(buttonPin) == HIGH) {
```

```
        state = STATE_VACIO;
      }

      if (digitalRead(pirPin) == HIGH) {
        contador_p++;
        aireando();
      }

      if ((currentTime - startTime).totalseconds() >= 15 && temperature
< 30) {
        startTime = currentTime;
        state = STATE_FINALIZADO;
      }
      break;

    case STATE_FINALIZADO:
      lcd.setCursor(0, 1);
      lcd.print("FINALIZADO          ");
      lcd.setCursor(0, 0);
      lcd.print("REINICIAR           ");
      contador_p = 0;
       if (digitalRead(buttonPin) == HIGH) {
        state = STATE_VACIO;
      }
      break;
  }
}
```